REMARKS

This Amendment is in response to the Office Action dated May 7, 2007, in which claims 1-20 were initially rejected. Applicants respectfully request reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

I. CLAIM AMENDMENTS

Independent claims 1, 10 and 13 are amended to clarify one or more elements of the claims, not to distinguish over any particular prior art reference. In particular, these claims are not amended to distinguish over Sharma et al., U.S. Patent No. 5,842,663. Rather, these claims are amended to provide a better context for the claim limitations.

Claims 11 and 14 are amended to be consistent with their respective independent claims.

II. PRESENT APPLICATION

A. **Claim 1**

Claim 1 is directed to a process of abstracting file paths to locations of a plurality of design files. The process includes inputting at least one description file that is located on a system and defines the design files in a first environment. The process parses a directory structure on the system to locate the description file and parses the description file to identify file paths to the description file and each of the design files defined by the description file. These file paths correspond to paths through the directory structure to locations of the description file and each of the design files. The process further generates an index correlating each description file and its respective file paths for the first environment.

B. **Example**

In one illustrative and non-limiting example, this process of abstracting file paths to locations of a plurality of design files allows the design files to be moved easily from one system to another system since the path names to the design files are not hard-coded.

Rather, the process can simply be executed again to re-generate the index, which correlates each description file and its respective file path.

As such, the process provides an automated technique to manage IC design file paths has the design files are applied to different environments and tools.

For example, a tool can be used to dynamically generate lists of the source and library files for each shell describing an aspect of a design that other tools will manipulate. The designer can use higher level "makefiles" that manipulate the design files for each shell. As a result, hard-coded and relative paths in shell script usage are eliminated.

In one example the description files provide a hierarchical list of the source or design files and RTL. The designer builds a description file for a portion of the design, which will reference design files for each of the circuit blocks in the module.

By separating the directory structure from the tool flow, greater flexibility is permitted of directory structures and separation removes the need for hard-coded and relative paths definitions for components. The use of the description files to identify design files (e.g. IC manufacture files and customer-created files) allows for the transportation of the entire design, or just a portion of it, without requiring the user/customer, to understand which files belong to which shell tasks and without requiring/customer to understand the directory structure.

II. CLAIM REJECTIONS UNDER §102

Claims 1-20 were rejected under §102(b) as being allegedly anticipated by Sharma et al., U.S. Patent No. 5,842,663.

A. **Claim 1**

1. Misinterpretation of "File Paths"

It appears from the Office Action that the Examiner may have confused the term "file paths" in the present claims with the term "datapaths" in Sharma et al.

A "file path" refers to the location of a file on a system, whereas a "datapath" in Sharma et al. refers a path along which a signal travels through one or more circuit elements of an integrated circuit.

The HDL circuit description of Sharma et al. therefore defines the structure of a circuit having a datapath within the circuit itself.

2. Sharma et al.

As described in column 1, lines 23-27, an object of the Sharma et al. patent is to "provide an ASIC synthesizer that synthesizes a circuit netlist from an HDL circuit description using a library of datapath circuit elements (i.e., circuit elements with formal HDL circuit descriptions) and a library of gate elements." The datapath library supports "the creation of customized components." (Column 1, lines 57-59). "When a parameterized HDL module is identified for use, parameter values are assigned to form an instance of the parameterized HDL module. This instance of the parameterized HDL module is then compiled into a netlist." (Column 2, lines 19-23).

With respect to claim 1, the Office Action suggests Sharma anticipates each and every element of claim 1 and refers to figure 16, HDL circuit description 28, parameterized HDL library modules 300A-300N, parser 32 and intermediate RTL 34.

However, these elements relate a datapath synthesizer, which accesses a library and assigns parameters to form specific implementations of parameterized HDL modules. (See, Abstract). The elements shown in figure 16 relate to operations performed on the design of the circuit itself, not parsing a directory structure on the system to locate the description file or parsing the description file to identify file paths to the description file and each of the design files, as recited in claim 1.

In contrast, parser 32 (shown in figure 1 and figure 16 of Sharma et al.), "translates the HDL (hardware description language) circuit description 28 into an intermediate register transfer level (RTL) structure 34 corresponding to the circuit." (Column 3, lines 43-47) (emphasis added). Thus, parser 32 has nothing to do with file paths in a directory structure on a system.

Further, Sharma et al. do not disclose the step of "generating an index correlating each description file and its respective file path for the first environment," as recited in claim 1.

The Office Action refers to the intermediate RTL description 34 and its related text of Sharma et al. However, the RTL relates to the circuit itself, not an index that correlates a description file and its respective file path. In other words, Sharma et al. do not suggest that the intermediate RTL description 34 includes an index that correlates a description file (that might identify such an intermediate RTL description 34) and is respective file path.

Sharma et al. therefore fails to anticipate the elements of independent claim 1 and the elements of dependent claims 2-12 either in original form or as amended above.

B. **Dependent Claims 2-9**

One or more of dependent claims 2-9 add further elements that are also not anticipated by Sharma et al.

In the rejection of each of the dependent claims, it appears that the Office Action continues to confuse the term "datapaths" (signal paths in the circuit itself) with "files paths" (paths to locations of files on a system).

For example, with respect to claim 2, Sharma et al. does not disclose the steps of defining a directory of description files defining file paths and parsing the directory. Rather, parser 32 of Sharma et al. parses the HDL description to assign values to circuit parameters in the HDL.

A further example of such confusion can be seen in dependent claim 9, which recites that, in combination with claim 8, the files paths in the index are concatenated with respective file paths in the second environment.

Page 42, lines 7-12 provides a non-limiting example in which, "The complete path is constructed to the form <full_path>/file.ext by combining the location of the .rmk file ($LSI_RI/sim/sve), shown at 52 in FIG. 7, with the value (testbench/Ucm.Tb.v), shown at 50 in FIG. 11, resulting in the complete path 54 shown in FIG. 14."

The Office Action refers Applicants' attention to Sharma, column 26, lines 50-55 which states that, "the compiled netlist can be combined with other netlist elements to form a complete netlist implementing the behavior of the HDL circuit description 28."

Thus, elements of one list are combined with elements of another list. But this has nothing to do with concatenating file paths.

C. **Claims 10-11**

Claims 10-11 are directed to a process of applying a plurality of design files in a hardware description language to a second environment.

As discussed above, Sharma et al. do not provide an index correlating a description file and its respective file path in a first environment, wherein the description file defines file paths to the design files in a first environment, as recited in claim 10.

D. **Claims 13-20**

Claims 13-20 relate to a computer useable medium having computer readable program code for performing a process similar to that discussed above with respect to claim 1-9. For similar reasons, claims 13-20 are not anticipated by Sharma et al.

Accordingly, Applicants respectfully request that the rejection of claims 1-20 under §102(b) based on Sharma et al. be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 12-2252.

Respectfully submitted,
WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/
David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB/tkj